

# Transmission Control Protocol

Leonardo H. Añez Vladimirovna\*

*Universidad Autónoma Gabriel René Moreno,  
Facultad de Ingeniería en Ciencias de la Computación y Telecomunicaciones,  
Santa Cruz de la Sierra, Bolivia*

20 de noviembre de 2019

El protocolo de control de transmisión (TCP) es uno de los protocolos mas importantes del conjunto de protocolos de internet. Es el protocolo mas utilizado para la transmisión de datos en redes de comunicación como internet.

## Características

- TCP es un protocolo confiable. El receptor siempre envía un **ACK** positivo o negativo sobre el paquete de datos remitente de modo que el remitente siempre tiene una pista clara sobre si el paquete de datos llegó al destino o si necesita reenviarlo.
- **Orientado a la conexión.** Se requiere una conexión entre dos puntos remotos antes de enviar datos reales.
- Proporciona un mecanismo de verificación y recuperación de errores.
- Proporciona comunicación extremo a extremo.
- Funciona en modo punto a punto. (Cliente/Servidor).

## Header

Source Port		Destination Port	
Sequence Number			
Acknowledgment Number			
Offset	Res.	Flags	Window
Checksum		Urgent Pointer	
Options			

- **Source Port(16Bits):** Identifica el puerto de origen del proceso de solicitud en el dispositivo emisor.
- **Destination Port(16Bits):** Identifica el puerto de destino del proceso de solicitud en el dispositivo receptor.
- **Sequence Number(32Bits):** número de secuencia de bytes de datos de un segmento en una sesión.
- **Acknowledgement Number(32Bits):** cuando se activa el indicador **ACK** este numero contiene el siguiente numero de secuencia del byte de datos esperado y funciona como **ACK** de recibo de los datos recibidos anteriormente.

---

\*Correo Electrónico: toborochi98@outlook.com

- **Data Offset(4Bits)**: este campo implica tanto el tamaño del encabezado TCP (palabra de 32 bits) como el desplazamiento de datos en el paquete actual en todo el segmento.
- **Reserved(3Bits)**: Reservado para uso futuro y todos están configurados en 0 de manera predeterminada.
- **Flags**: indicadores de bits que indican como debe intepretar.
  - **SYN**: Es usado durante el establecimiento de la conexión.
  - **FIN**: Es usado durante la liberación de la conexión.
  - **RST**: Se usa en caso de problemas o cuando un segmento invalido es recibido.
  - **ACK**: Cuando se lo establece, indica que el campo de reconocimiento contiene un número valido. De lo contrario, el receptor debe ignorar el contenido del campo de reconocimiento.
  - **URG**: Es usado junto con el Urgent Pointer.
  - **PSH**: Cuando se establece, es una solicitud a la estación receptora para empujar datos (tan pronto como llegue) a la aplicación receptora sin almacenarlo en el buffer.
  - **CWB**: Cuando un host recibe un paquete con el conjunto de bits ECE, establece la reducción de congestión de ventana, para confirmar recibe ECE.
  - **ECE**: Tiene dos significados:
    - Si el bit **SYN** es 0, entonces **ECE** significa que el paquete IP tiene establecido su bit **CE** (Congestion).
    - Si el bit **SYN** se establece en 1, **ECE** significa que el dispositivo es compatible.
- **Window**: Este campo se usa para el control de flujo entre dos estaciones o indica la cantidad de buffer (bytes) que el receptor ha asignado para un segmento, es decir **cuantos datos espera el receptor**.
- **Checksum**: La verificación.
- **Urgent Pointer**: apunta al byte de datos urgentes si el indicador esta en 1.
- **Options**: opciones adicionales que no están cubiertos por el encabezado regular.

## Addressing

La comunicación TCP entre dos hosts remotos se realiza mediante numeros de puerto, van desde 0 a 65535:

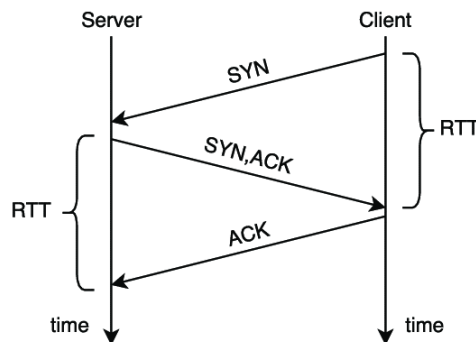
- System Ports(0-1023)
- User Ports(1024-49151)
- Private/Dynamic Ports (49152-65535)

## Gestión de la Conexión

Funciona en el modelo Servidor/Cliente. El cliente inicia la conexión y el servidor la acepta o la rechaza. El protocolo de enlace de 3 vías (3 Ways Handshake) se usa para la administración de la conexión.

### Establecimiento

El cliente inicia la conexión y envía el segmento con un número de secuencia. El servidor lo confirma con su propio número de secuencia y **ACK** del segmento del cliente que es uno mas que el numero de secuencia del cliente. El cliente después de recibir **ACK** de su segmento envía una señal de recibo de la respuesta del servidor.



### Liberación

Tanto el servidor como el cliente pueden enviar un segmento con el indicador **FIN** establecido en 1. Cuando el extremo receptor responde con **ACK FIN**, se cierra y libera.

## Errores y Flujos

Utiliza numeros de puerto para saber que proceso de aplicación necesita para transferir el segmento de datos. Junto con eso, utiliza un número de secuencia para sincronizar con el host remoto. Todos los segmentos de datos se envían y reciben con números de secuencia. El remitente sabe que ultimo segmento de datos recibió el receptor cuando recibe **ACK**. El receptor conoce el ultimo segmento enviado por el remitente al referirse al número de secuencia del recibido.

## Control de Congestión

Cuando llega una gran cantidad de datos al sistema que no es capaz de manejarlos se produce congestión. Controla esto por la ventana. Establece un tamaño de ventana que le dice al otro extremo cuantos segmentos de datos enviar, se usan tres algoritmos:

- Incremento Aditivo, Decremento Multiplicativo
- Slow Start
- Timeout React